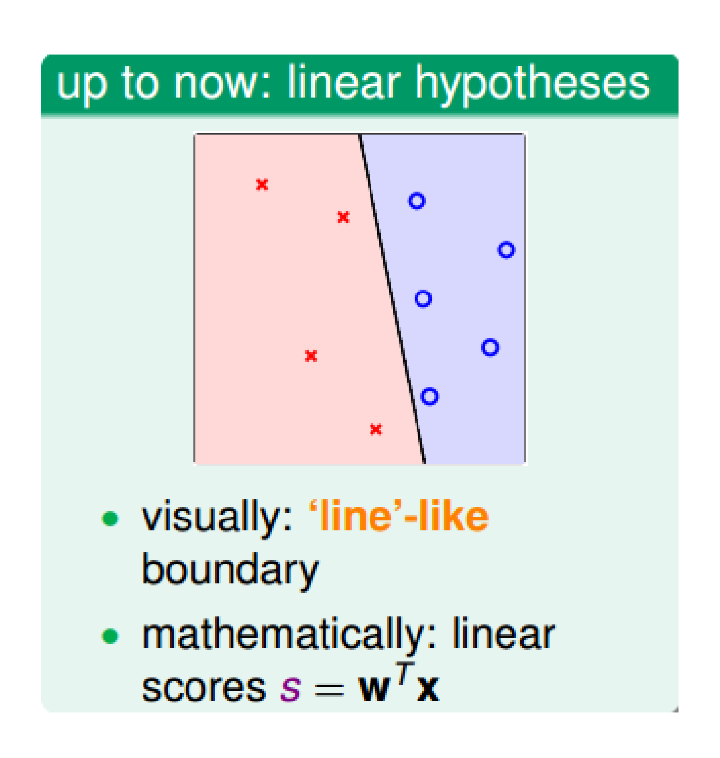
林轩田《机器学习基石》课程笔记12 -- Nonlinear Transformation

作者：红色石头　　公众号：AI有道（id：redstonewill）

上一节课，我们介绍了分类问题的三种线性模型，可以用来解决binary classification和multiclass classification问题。本节课主要介绍非线性的模型来解决分类问题。

一、Quadratic Hypothesis

之前介绍的线性模型，在2D平面上是一条直线，在3D空间中是一个平面。数学上，我们用线性得分函数s来表示：$s = w^T x$。其中，x为特征值向量，w为权重，s是线性的。



线性模型的优点就是，它的VC Dimension比较小，保证了$E_{in} \approx E_{out}$。但是缺点也很明显，对某些非线性问题，可能会造成E_{in}很大，虽然$E_{in} \approx E_{out}$，但是也造成E_{out}很大，分类效果不佳。

but limited . . .

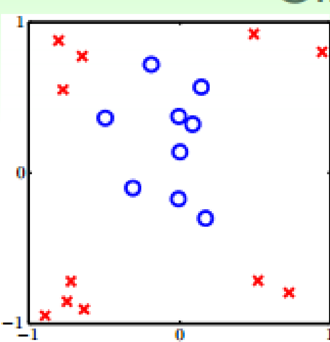
- theoretically: d_{VC} under control :-)
- practically: on some $\mathcal{D}$, large E_{in} for every line :-(

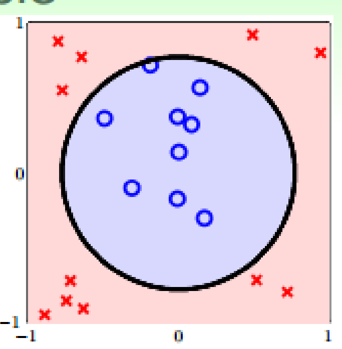
为了解决线性模型的缺点，我们可以使用非线性模型来进行分类。例如数据集D不是线性可分的，而是圆形可分的，圆形内部是正类，外面是负类。假设它的hypotheses可以写成：

$$h_{SEP}(x) = sign(-x_1^2 - x_2^2 + 0.6)$$

基于这种非线性思想，我们之前讨论的PLA、Regression问题都可以有非线性的形式进行求解。

Circular Separable





- $\mathcal{D}$ not linear separable
- but **circular separable** by a circle of radius $\sqrt{0.6}$ centered at origin:

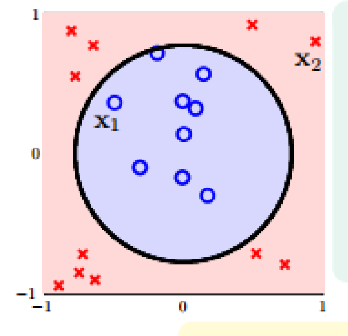
$$h_{\text{SEP}}(\mathbf{x}) = \text{sign}\left(-x_1^2 - x_2^2 + 0.6\right)$$

re-derive **Circular**-PLA, **Circular**-Regression, blahblah . . . all over again? :-)

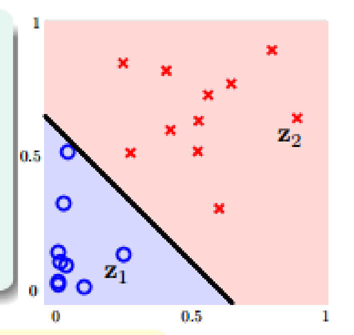
下面介绍如何设计这些非线性模型的演算法。还是上面介绍的平面圆形分类例子，它的h(x)的权重w0=0.6，w1=-1，w2=-1，但是h(x)的特征不是线性模型的$(1,x_1,x_2)$，而是$(1,x_1^2,x_2^2)$。我们令$z_0=1$，$z_1=x_1^2$，$z_2=x_2^2$，那么，h(x)变成：

$$h(x)=sign(\breve{w}_0\cdot z_0+\breve{w}_1\cdot z_1+\breve{w}_2\cdot z_2)=sign(0.6\cdot z_0-1\cdot z_1-1\cdot z_2)=sign(\breve{w}^Tz)$$

这种$x_n\rightarrow z_n$的转换可以看成是x空间的点映射到z空间中去，而在z域中，可以用一条直线进行分类，也就是从x空间的圆形可分映射到z空间的线性可分。z域中的直线对应于x域中的圆形。因此，我们把$x_n\rightarrow z_n$这个过程称之为特征转换（Feature Transform）。通过这种特征转换，可以将非线性模型转换为另一个域中的线性模型。



- $\{(\mathbf{x}_n, y_n)\}$ circular separable $\Longrightarrow \{(\mathbf{z}_n, y_n)\}$ linear separable
- $\mathbf{x} \in \mathcal{X} \overset{\Phi}{\longmapsto} \mathbf{z} \in \mathcal{Z}$: **(nonlinear) feature transform Φ**



circular separable in $\mathcal{X} \Longrightarrow$ linear separable in $\mathcal{Z}$
vice versa?

已知x域中圆形可分在z域中是线性可分的，那么反过来，如果在z域中线性可分，是否在x域中一定是圆形可分的呢？答案是否定的。由于权重向量w取值不同，x域中的hypothesis可能是圆形、椭圆、双曲线等等多种情况。

$$(z_0, z_1, z_2) = \mathbf{z} = \mathbf{\Phi}(\mathbf{x}) = (1, x_1^2, x_2^2)$$
$$h(\mathbf{x}) = \tilde{h}(\mathbf{z}) = \text{sign}\left(\tilde{\mathbf{w}}^T \mathbf{\Phi}(\mathbf{x})\right) = \text{sign}\left(\tilde{w}_0 + \tilde{w}_1 x_1^2 + \tilde{w}_2 x_2^2\right)$$

$\tilde{\mathbf{w}} = (\tilde{w}_0, \tilde{w}_1, \tilde{w}_2)$

- $(0.6, -1, -1)$: circle ($\circ$ inside)
- $(-0.6, +1, +1)$: circle ($\circ$ outside)
- $(0.6, -1, -2)$: ellipse
- $(0.6, -1, +2)$: hyperbola
- $(0.6, +1, +2)$: constant $\circ$:-)

目前讨论的x域中的圆形都是圆心过原点的，对于圆心不过原点的一般情况，$x_n \to z_n$映射公式包含的所有项为：

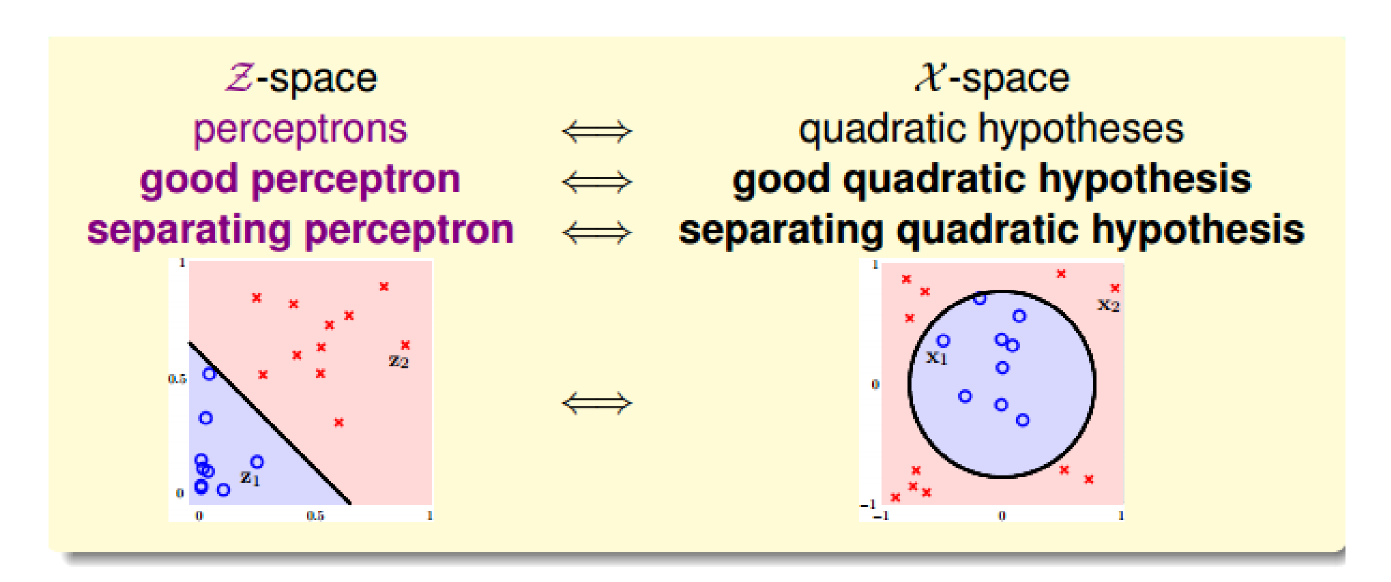
$$\Phi_2(x) = (1, x_1, x_2, x_1^2, x_1 x_2, x_2^2)$$

也就是说，对于二次hypothesis，它包含二次项、一次项和常数项1，z域中每一条线对应x域中的某二次曲线的分类方式，也许是圆，也许是椭圆，也许是双曲线等等。那么z域中的hypothesis可以写成：

$$\mathcal{H}_{\Phi_2} = \left\{h(\mathbf{x}) : h(\mathbf{x}) = \tilde{h}(\mathbf{\Phi}_2(\mathbf{x})) \text{ for some linear } \tilde{h} \text{ on } \mathcal{Z}\right\}$$

二、Nonlinear Transform

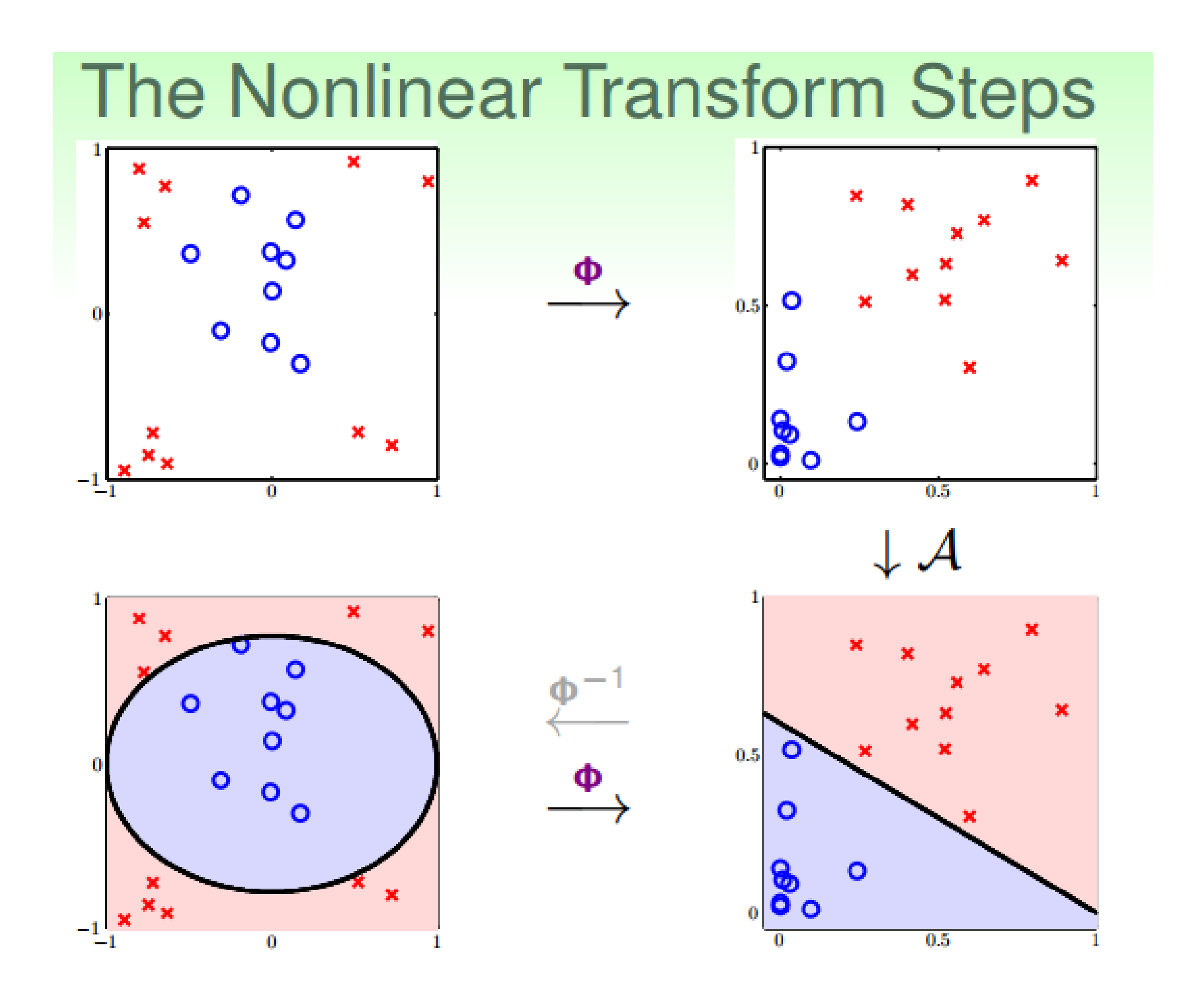
上一部分我们定义了什么了二次hypothesis，那么这部分将介绍如何设计一个好的二次hypothesis来达到良好的分类效果。那么目标就是在z域中设计一个最佳的分类线。



- want: get **good perceptron** in $\mathcal{Z}$-space
- known: get **good perceptron** in $\mathcal{X}$-space with data $\{(\mathbf{x}_n, y_n)\}$

todo: get **good perceptron** in $\mathcal{Z}$-space with data $\{(\mathbf{z}_n = \mathbf{\Phi}_2(\mathbf{x}_n), y_n)\}$

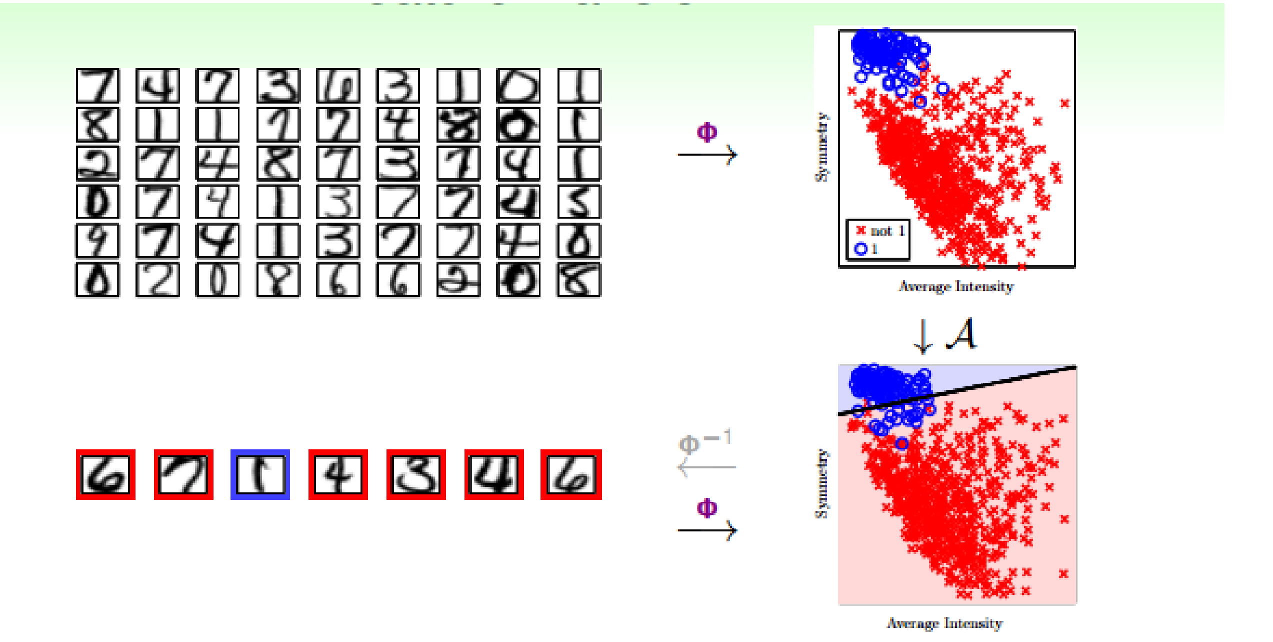
其实，做法很简单，利用映射变换的思想，通过映射关系，把x域中的最高阶二次的多项式转换为z域中的一次向量，也就是从quardratic hypothesis转换成了perceptrons问题。用z值代替x多项式，其中向量z的个数与x域中x多项式的个数一致（包含常数项）。这样就可以在z域中利用线性分类模型进行分类训练。训练好的线性模型之后，再将z替换为x的多项式就可以了。具体过程如下：



整个过程就是通过映射关系，换个空间去做线性分类，重点包括两个：

- 特征转换
- 训练线性模型

其实，我们以前处理机器学习问题的时候，已经做过类似的特征变换了。比如数字识别问题，我们从原始的像素值特征转换为一些实际的concrete特征，比如密度、对称性等等，这也用到了feature transform的思想。



not new, not just polynomial:

raw (pixels) $\xrightarrow{\text{domain knowledge}}$ concrete (intensity, symmetry)

三、Price of Nonlinear Transform

若x特征维度是d维的，也就是包含d个特征，那么二次多项式个数，即z域特征维度是：

$$\check{d} = 1 + C_d^1 + C_d^2 + d = \frac{d(d+3)}{2} + 1$$

如果x特征维度是2维的，即(x_1, x_2)，那么它的二次多项式为$(1, x_1, x_2, x_1^2, x_1x_2, x_2^2)$，有6个。

现在，如果阶数更高，假设阶数为Q，那么对于x特征维度是d维的，它的z域特征维度为：

$$\check{d} = C_{Q+d}^{Q} = C_{Q+d}^{d} = O(Q^d)$$

由上式可以看出，计算z域特征维度个数的时间复杂度是Q的d次方，随着Q和d的增大，计算量会变得很大。同时，空间复杂度也大。也就是说，这种特征变换的一个代价是计算的时间、空间复杂度都比较大。

Q-th order polynomial transform: $\Phi_Q(\mathbf{x}) = \big(1, x_1, x_2, \ldots, x_d, x_1^2, x_1 x_2, \ldots, x_d^2, \ldots, x_1^Q, x_1^{Q-1} x_2, \ldots, x_d^Q \big)$

$\underbrace{1}_{\tilde{w}_0} + \underbrace{\tilde{d}}_{\text{others}}$ dimensions

= # ways of $\leq Q$-combination from d kinds with repetitions

$= \binom{Q+d}{Q} = \binom{Q+d}{d} = O(Q^d)$

= efforts needed for computing/storing $\mathbf{z} = \Phi_Q(\mathbf{x})$ and $\tilde{\mathbf{w}}$

Q large $\Longrightarrow$ **difficult to compute/store**

另一方面，z域中特征个数随着Q和d增加变得很大，同时权重w也会增大，即自由度增加，VC Dimension增大。令z域中的特征维度是$1+\breve{d}$，则在在域中，任何$\breve{d}+2$的输入都不能被shattered；同样，在x域中，任何$\breve{d}+2$的输入也不能被shattered。$\breve{d}+1$是VC Dimension的上界，如果$\breve{d}+1$很大的时候，相应的VC Dimension就会很大。根据之前章节课程的讨论，VC Dimension过大，模型的泛化能力会比较差。

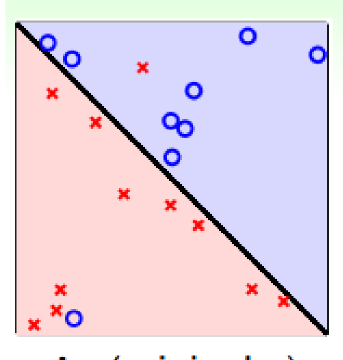
$\underbrace{1}_{\tilde{w}_0} + \underbrace{\tilde{d}}_{\text{others}}$ dimensions $= O(Q^d)$

- number of free parameters $\tilde{w}_i = \tilde{d} + 1 \approx d_{\text{VC}}(\mathcal{H}_{\Phi_Q})$
- $d_{\text{VC}}(\mathcal{H}_{\Phi_Q}) \leq \tilde{d} + 1$, why?

  any $\tilde{d}+2$ inputs not shattered in $\mathcal{Z}$
  $\Longrightarrow$ any $\tilde{d}+2$ inputs not shattered in $\mathcal{X}$

Q large $\Longrightarrow$ **large d_{VC}**

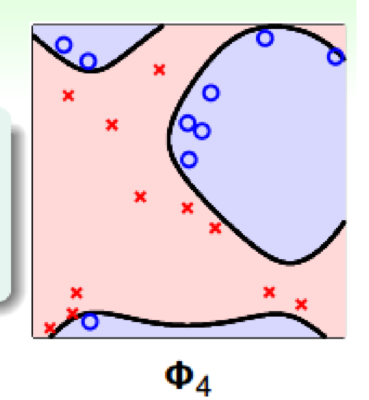
下面通过一个例子来解释为什么VC Dimension过大，会造成不好的分类效果：



Φ_1 (original $\mathbf{x}$)

which one do you prefer? :-)

- Φ_1 'visually' preferred
- Φ_4: $E_{\text{in}}(g) = 0$ but overkill



Φ_4

1. can we make sure that $E_{\text{out}}(g)$ is close enough to $E_{\text{in}}(g)$?
2. can we make $E_{\text{in}}(g)$ small enough?

trade-off:

$\tilde{d}$ (Q)	①	②
higher	:-(	**:-D**
lower	**:-D**	:-(

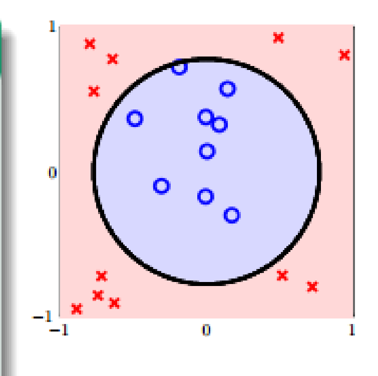
上图中，左边是用直线进行线性分类，有部分点分类错误；右边是用四次曲线进行非线性分类，所有点都分类正确，那么哪一个分类效果好呢？单从平面上这些训练数据来看，四次曲线的分类效果更好，但是四次曲线模型很容易带来过拟合的问题，虽然它的E_{in}比较小，从泛化能力上来说，还是左边的分类器更好一些。也就是说VC Dimension过大会带来过拟合问题，$\check{d}+1$不能太大了。

那么如何选择合适的Q，来保证不会出现过拟合问题，使模型的泛化能力强呢？一般情况下，为了尽量减少特征自由度，我们会根据训练样本的分布情况，人为地减少、省略一些项。但是，这种人为地删减特征会带来一些“自我分析”代价，虽然对训练样本分类效果好，但是对训练样本外的样本，不一定效果好。所以，一般情况下，还是要保存所有的多项式特征，避免对训练样本的人为选择。

Visualize $\mathcal{X} = \mathbb{R}^2$

- full Φ_2: $\mathbf{z} = (1, x_1, x_2, x_1^2, x_1x_2, x_2^2)$, $d_{\text{VC}} = 6$
- or $\mathbf{z} = (1, x_1^2, x_2^2)$, $d_{\text{VC}} = 3$, **after visualizing**?
- or better $\mathbf{z} = (1, x_1^2 + x_2^2)$, $d_{\text{VC}} = 2$?
- or even better $\mathbf{z} = (\text{sign}(0.6 - x_1^2 - x_2^2))$?

—careful about **your brain's 'model complexity'**



for VC-safety, Φ shall be decided **without 'peeking'** data

四、Structured Hypothesis Sets

下面，我们讨论一下从x域到z域的多项式变换。首先，如果特征维度只有1维的话，那么变换多项式只有常数项：

$$\Phi_0(x) = (1)$$

如果特征维度是两维的，变换多项式包含了一维的$\Phi_0(x)$：

$$\Phi_1(x) = (\Phi_0(x), x_1, x_2, \ldots, x_d)$$

如果特征维度是三维的，变换多项式包含了二维的$\Phi_1(x)$：

$$\Phi_2(x) = (\Phi_1(x), x_1^2, x_1 x_2, \ldots, x_d^2)$$

以此类推，如果特征维度是Q次，那么它的变换多项式为：

$$\Phi_Q(x) = (\Phi_{Q-1}(x), x_1^Q, x_1^{Q-1} x_2, \cdots, x_d^Q)$$

那么对于不同阶次构成的hypothesis有如下关系：

$$H_{\Phi_0} \subset H_{\Phi_1} \subset H_{\Phi_2} \subset \cdots \subset H_{\Phi_Q}$$

我们把这种结构叫做Structured Hypothesis Sets：

$$\begin{array}{ccccccccccc} \mathcal{H}_{\Phi_0} & \subset & \mathcal{H}_{\Phi_1} & \subset & \mathcal{H}_{\Phi_2} & \subset & \mathcal{H}_{\Phi_3} & \subset & \cdots & \subset & \mathcal{H}_{\Phi_Q} \\ \| & & \| & & \| & & \| & & & & \| \\ \mathcal{H}_0 & & \mathcal{H}_1 & & \mathcal{H}_2 & & \mathcal{H}_3 & & \cdots & & \mathcal{H}_Q \end{array}$$

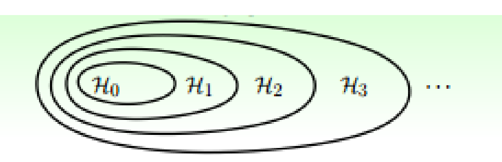
$\mathcal{H}_0$ $\mathcal{H}_1$ $\mathcal{H}_2$ $\mathcal{H}_3$ $\cdots$

那么对于这种Structured Hypothesis Sets，它们的VC Dimension满足下列关系：

$$d_{VC}(H_0) \leq d_{VC}(H_1) \leq d_{VC}(H_2) \leq \cdots \leq d_{VC}(H_Q)$$

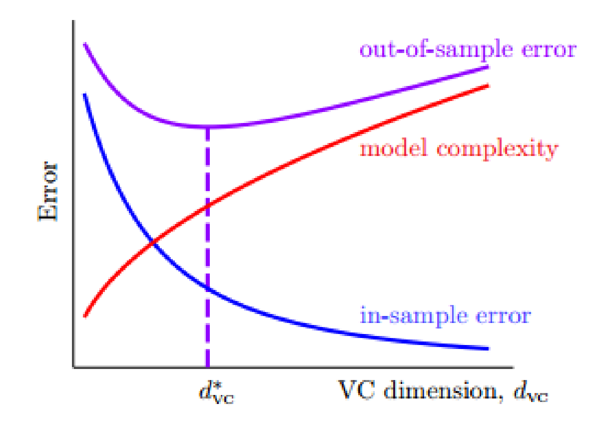
它的E_{in}满足下列关系：

$$E_{in}(g_0) \geq E_{in}(g_1) \geq E_{in}(g_2) \geq \cdots \geq E_{in}(g_Q)$$



Let $g_i = \text{argmin}_{h \in \mathcal{H}_i} E_{\text{in}}(h)$:

$$
\begin{array}{ccccccccc}
\mathcal{H}_0 & \subset & \mathcal{H}_1 & \subset & \mathcal{H}_2 & \subset & \mathcal{H}_3 & \subset & \cdots \\
d_{\text{vc}}(\mathcal{H}_0) & \leq & d_{\text{vc}}(\mathcal{H}_1) & \leq & d_{\text{vc}}(\mathcal{H}_2) & \leq & d_{\text{vc}}(\mathcal{H}_3) & \leq & \cdots \\
E_{\text{in}}(g_0) & \geq & E_{\text{in}}(g_1) & \geq & E_{\text{in}}(g_2) & \geq & E_{\text{in}}(g_3) & \geq & \cdots
\end{array}
$$



use $\mathcal{H}_{1126}$ won't be good! :-(

从上图中也可以看到，随着变换多项式的阶数增大，虽然E_{in}逐渐减小，但是model complexity会逐渐增大，造成E_{out}很大，所以阶数不能太高。

那么，如果选择的阶数很大，确实能使E_{in}接近于0，但是泛化能力通常很差，我们把这种情况叫做tempting sin。所以，一般最合适的做法是先从低阶开始，如先选择一阶hypothesis，看看E_{in}是否很小，如果E_{in}足够小的话就选择一阶，如果E_{in}大的话，再逐渐增加阶数，直到满足要求为止。也就是说，尽量选择低阶的hypothes，这样才能得到较强的泛化能力。

- tempting sin: use $\mathcal{H}_{1126}$, low $E_{\text{in}}(g_{1126})$ to fool your boss —**really? :-(a dangerous path of no return**
- safe route: $\mathcal{H}_1$ first
  - if $E_{\text{in}}(g_1)$ good enough, **live happily thereafter :-)**
  - otherwise, move right of the curve **with nothing lost except 'wasted' computation**

linear model first:
simple, efficient, **safe**, **and workable!**

五、总结

这节课主要介绍了非线性分类模型，通过非线性变换，将非线性模型映射到另一个空间，转换为线性模型，再来进行线性分类。本节课完整介绍了非线性变换的整体流程，以及非线性变换

可能会带来的一些问题：时间复杂度和空间复杂度的增加。最后介绍了在要付出代价的情况下，使用非线性变换的最安全的做法，尽可能使用简单的模型，而不是模型越复杂越好。

注明：

文章中所有的图片均来自台湾大学林轩田《机器学习基石》课程